

Appl. No.: 10/C39,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) In a hierarchical classification system including a classification tree comprising a plurality of traffic classes, wherein at least one traffic class of the plurality of traffic classes corresponds to a first classification type and at least one other traffic class in the plurality of traffic classes corresponds to a second classification type, wherein the classification tree further comprises a cache storing a data structure representing a cacheable portion of the classification tree, wherein the data structure corresponds to the at least one traffic class of the first classification type ~~performing caching~~, a method comprising:

walking ~~[[a]] the classification tree in a hierarchical classification system to~~ determine whether an incoming flow matches a traffic class in the classification tree; and

upon encountering the cacheable portion of the classification tree, performing a lookup on [[a]] the cache storing a data structure of a plurality of classes of one classification type to compare the incoming flow with a plurality of the at least one traffic class of the first classification type ~~classes at the same time to determine whether~~

Appl. No.: 10/039,992

Amdt. Dated October 26, 2005

Response to Office Action of June 28, 2005

the incoming flow matches one of the ~~plurality of classes~~ at least one traffic class.

2. (original) The method defined in Claim 1 wherein the data structure comprises a hash table.

3. (currently amended) The method defined in Claim 1 further comprising returning a class pointer indicative of user programming information that has been assigned to a traffic class if the incoming flow matches the traffic class.

4. (currently amended) The method defined in Claim 1 further characterized by performing a walk through the plurality of traffic classes in the classification tree if a determination that the traffic class in the at least one ~~plurality of traffic class~~ classes is not known as a result of performing the cache lookup.

5. (currently amended) The method defined in Claim 4 wherein if the ~~class of the~~ incoming flow packet hits the cache, then further comprising:

~~storing the class in the cache; and~~

returning a result indicating the incoming flow class was in the cache.

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

6. (currently amended) The method defined in Claim 4 wherein if the ~~class of the~~ incoming flow packet does not hit hits the cache and no traffic class is identified, further comprising:

~~marking the cache result as indicating that the class is not in the cache; and~~
continuing to walk the classification tree from a location in the classification tree immediately after the end of the cacheable portion of the classification tree represented in the cache.

7. (currently amended) An apparatus for use in a hierarchical classification system performing caching, the apparatus comprising:

a memory storing a classification tree comprising a plurality of traffic classes, wherein at lease one traffic class in the plurality of traffic classes corresponds to a first classification type and at least one other traffic class in the plurality of traffic classes corresponds to a second classification type;

a cache operative to store a cacheable portion of the classification tree, wherein the cacheable portion of the classification tree corresponds to at least one traffic class in the plurality of traffic classes of the first classification type; and

a classification engine coupled to the memory to walk the classification tree as ~~part of a hierarchical classification system~~ to determine whether an incoming flow

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

matches a traffic class in the classification tree, and, when the cacheable portion of the classification tree is encountered, to perform a lookup on the cache storing a data structure of a plurality of classes of one classification type to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the at least one traffic classes of the first classification type plurality of classes.

8. (original) The apparatus defined in Claim 7 wherein the data structure comprises a hash table.

9. (currently amended) The apparatus defined in Claim 7 wherein the classification engine returns a class pointer indicative of user programming information that has been assigned to a traffic class if the incoming flow matches the traffic class.

10. (currently amended) The apparatus defined in Claim 7 wherein the classification engine performs a walk through the cacheable portion of the classification tree plurality of classes if a determination that the traffic class corresponding to the incoming flow in the plurality of classes is not known as a result of performing the cache lookup.

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

11. (currently amended) The apparatus defined in Claim 10 wherein if the ~~class of~~ the incoming flow packet hits matches a traffic class corresponding to a cacheable portion of the classification tree ~~the cache~~, then the classification engine:

creates an entry in the data structure including an attribute of the incoming flow defined by the first classification type and the matching traffic class; and

~~stores the traffic class in the cache; and~~

returns a result indicating the matching traffic class ~~was in the cache~~.

12. (currently amended) The apparatus defined in Claim 10 wherein if the ~~class of~~ the incoming flow packet does not match a traffic class in the cacheable portion of the classification tree ~~hit the cache~~, then the classification engine:

creates an entry in the data structure including an attribute of the incoming flow defined by the first classification type and an indication that marks the cache result as ~~indicating that the class is not in the cache; and~~

continues to walk the classification tree from a location in the tree immediately after the end of the cacheable portion of the classification tree represented in the cache.

13. (currently amended) An apparatus for use in a hierarchical classification

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

system performing caching, the apparatus comprising:

a memory storing a classification tree comprising a plurality of traffic classes,
wherein at lease one traffic class in the plurality of traffic classes corresponds to a first
classification type and at least one other traffic class in the plurality of traffic classes
corresponds to a second classification type;

a cache operative to store a cacheable portion of the classification tree, wherein
the cacheable portion of the classification tree corresponds to at least one traffic class in
the plurality of traffic classes of the first classification type; and

a classification engine operative to:

determine, relative to the first classification type, whether the cache
contains an entry matching an incoming flow;

conditionally walk, if no entry is found in the cache, the cacheable portion
of the classification tree to match the incoming flow to the at least one traffic class, and
then:

create, if the incoming flow does not match one of the at least one traffic
class in the cacheable portion of the classification tree, an entry in the cache indicating
that the incoming flow, relative to the first classification type, does not match a traffic
class in the cacheable portion; or

create, if the incoming flow hits one of the at least one traffic class in the

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

cacheable portion of the classification tree, an entry in the cache, based on the first classification type, corresponding to the incoming flow and the matching traffic class.

~~_____ means for walking a classification tree in a hierarchical classification system to determine whether an incoming flow matches a class in the classification tree; and~~

~~_____ means for performing a lookup on a cache storing a data structure of a plurality of classes of one classification type to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.~~

14. (canceled)

15. (currently amended) In a hierarchical classification system performing caching, a method for classifying a flow comprising:

ordering a classification tree including a plurality traffic classes, wherein at least one traffic class corresponds to a first classification type, and wherein at least another traffic class corresponds to a second classification type of classification types by grouping at least a portion of the traffic classes of each classification type of classification together in the classification tree;

creating a first data structure, representing a first cacheable portion of the

Appl. No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

classification tree, of a plurality of traffic classes of ~~one~~ the first classification type to facilitate determination of ~~determine~~ whether the flow matches any traffic classes ~~items~~ in the first data structure;

~~_____~~ creating a second data structure, representing a second cacheable portion of the classification tree, of a plurality of traffic classes of the second classification type to facilitate determination of whether the flow matches any traffic classes in the second data structure.

~~_____ performing tree walking of the classification tree; and~~

~~_____ performing a lookup on a cache storing the data structure to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.~~

16. (currently amended) The method defined in Claim 15 wherein the first data structure comprises a hash table.

17. (currently amended) The method defined in Claim 15 wherein the first classification type ~~types~~ in the classification tree includes network addresses.

18. (currently amended) The method defined in Claim 17 wherein the second

Appl No.: 10/039,992
Amdt. Dated October 26, 2005
Response to Office Action of June 28, 2005

classification ~~type types~~ in the classification tree includes one or more network services.

19. (currently amended) The method defined in Claim 18 wherein the second classification ~~type types~~ in the classification tree further include on or more ports.

20. (currently amended) The method defined in Claim 17 wherein the network addresses comprise Internet Protocol (IP) addresses.

21. (canceled)

22. (new) The method of claim 1 further comprising
conditionally walking, if no entry is found in the data structure, the
cacheable portion of the classification tree to match the incoming flow to the at least
one traffic class, and then:
creating, if the incoming flow does not match one of the at least one traffic
class in the cacheable portion of the classification tree, an entry in the data structure
indicating that the incoming flow, relative to the first classification type, does not match
a traffic class in the cacheable portion; or

creating, if the incoming flow matches one of the at least one traffic class

Appl. No.: 10/039,992

Am.dt. Dated October 26, 2005

Response to Office Action of June 28, 2005

in the cacheable portion of the classification tree, an entry in the data structure, based on the first classification type, corresponding to the incoming flow and the matching traffic class.

23. (new) The apparatus of claim 13 wherein the cache comprises a hash table.